

Validació automàtica d'imatges de comptadors de gas

Aitor Lagunas Fernández

Resum— Aquest projecte consisteix en la validació de números basada en el reconeixement d'imatges, en concret, imatges de comptadors de gas on poden haver-hi diferents models. S'utilitzen uns descriptors de contorn HOG per les imatges, de manera que s'aconsegueix, mitjançant un entrenament, un model. Aquest model l'obtenim gràcies a una representació no supervisada de les imatges on es pot distingir el número de sèrie de cada comptador.

Gràcies al mètode del SVM obtenim una classificació passant-li unes imatges de test on ens diu si són les imatges correctes o no. Finalment aconseguim un threshold (%) que ens indica quantes imatges positives i negatives ens ha detectat bé.

Aquestes proves s'han realitzat amb imatges sintètiques, és a dir, creades manualment, i imatges reals.

Paraules clau— HOG (Histogram of Oriented Gradients), SVM (Support Vector Machine), Matlab, Train, Model, Test, imatges sintètiques, imatges reals, Threshold

Abstract— This project involves the validation of numbers based on image recognition, namely gas meters where images can have different models. HOG descriptors are used for images. A model is achieved through training. This model we get thanks to unsupervised images where you can distinguish the serial number of each gas meter.

With the SVM method we obtain a classification passing test images which tell us whether or not images are correct.

Finally we get a threshold (%) that indicates how many positive and negative images has detected well. These tests were conducted with synthetic images, ie, manually created and real images.

Index Terms— HOG (Histogram of Oriented Gradients), SVM (Support Vector Machine), Matlab, Train, Model, Test, synthetic images, and real images, Threshold



1 INTRODUCCIÓ

A VUI dia, una bona manera d'estalviar temps i esforç a l'hora de validar el número de sèrie de cada comptador de gas que hi ha en totes les cases, pot ser la d'enviar una foto del teu comptador a "l'empresa" i que ells mitjançant un programa et validin el número. Doncs bé, aquest és l'objectiu principal del projecte: Identificar i validar els números de sèrie. Per a fer-ho, s'utilitza un OCR (Reconeixement òptic de caràcters) i el programa Matlab.

Un dels obstacles que es poden trobar és que hi ha diferents models de comptadors, així doncs, el número de sèrie es pot posicionar en diferents llocs de la imatge. Altres problemes que s'han donat, per exemple, són:

- Diferents tamanyes del números.
- Fotografia amb soroll.
- Tipografia dels números.
- Imatges mal enfocades o amb números tallats.

En les següents imatges es poden veure diferents comptadors de gas.



- E-mail de contacte: lagunas.aitor@gmail.com
- Menció realitzada: Enginyeria Informàtica de Computació.
- Treball tutoritzat per: Ernest Valveny (Departament de Ciències de la Computació)
- Curs 2014/15

Actualment es té un programa on es reconeix número a número i es valida per separat. Doncs bé, el que es vol és saber si és viable poder reconèixer tot el número en conjunt i validar-lo. És una alternativa que es planteja en el projecte, però continuant amb el model per poder decidir si el número que tenim a la imatge és el que hem introduït nosaltres. El projecte del qual fem la referència és [4].

Aquest document contindrà la metodologia utilitzada per fer els mètodes HOG i SVM, les tècniques utilitzades per poder obtenir el el model a partir de les imatges de train i el threshold a partir de les imatges de test, i finalment els resultats obtinguts amb les seves pertinents conclusions.

2 ESTAT DE L'ART

El projecte es basa en la segmentació de la imatge i l'aplicació del Reconeixement Òptic de Caràcters. L'OCR (referència [2]) és una tecnologia que permet convertir diferents tipus de documents, com per exemple, documents en paper escanejats, arxius PDF o imatges capturades per una càmera digital, en dades amb opció a cerca i funcionalitat a editar.

Es crea una finestra lliscant (referència [7]) amb una mida fixe per raons d'eficiència, ja que el número de sèrie sempre tindrà 8 dígit. Aquesta finestra va passant per tota la imatge i gràcies a uns descriptors HOG es pot comparar i validar el número introduït. Les imatges estan representades amb aquests descriptors.

L' Histograma de gradients orientats (HOG) (referència [13]) és un descriptor característica utilitzat per detectar objectes en visió per computador i processament d'imatges. La tècnica descriptor HOG compte les ocurrences de l'orientació del gradient en porcions localitzades d'una imatge, finestra de detecció, o regió d'interès (ROI).

L'aplicació de l'algorisme descriptor HOG és el següent:

- Dividir la imatge en petites regions i per cada regió calcular el gradient.
- Discretitzar cada regió d'acord amb l'orientació del gradient.
- S'agrupen les regions en un bloc i es realitza la normalització d'histogrames.
- Un grup normalitzat d'histogrames representa l'histograma bloc. El conjunt d'aquests histogrames de blocs representa el descriptor.

Aquestes imatges tenen un problema difícil per als descriptors perquè han de fer front a moltes fonts de variabilitat, com deformacions, diferents estils, etc. D'altra banda, en una gran escala i escenari de segmentació lliure, és obligatori l'ús de descriptors que són alhora ràpids per calcular i comparar, i que s'integren amb una finestra lliscant.

La següent tècnica a utilitzar és el Support Vector Machine, a partir d'ara, SVM (referència [6]). És un algoritme d'aprenentatge supervisat basat en la classificació i la regressió. Gràcies al conjunt d'entrenament es construeix un model on s'etiqueten les diferents classes de les imatges. Quan les noves mostres es comparen amb el model, el mètode les classifica en una classe o una altra.

3 METODOLOGIA

3.1 Adaptació

Com que de imatges positives només podem treure'n una per número de sèrie, el que s'ha fet és crear les imatges sintètiques.

La metodologia que s'utilitzarà en aquest projecte es basa en les diferents formes de generar imatges positives i negatives.

Es realitzaran diferents proves i després seran avaluades i comparades, podent així, treure'n una conclusió per determinar si són metodologies viables o no.

Les proves seran:

TRAIN		TEST	
POSITIVES	NEGATIVES	POSITIVES	NEGATIVES
S	S	S	S
S	S	R	R
S	R	R	R

S = Imatges sintètiques

R = Imatges reals

Primer hem generat les imatges sintètiques per fer tant el train com el test. La següent prova ha sigut incorporant les imatges reals en el test, però només podent utilitzar-ne una de positiva. Per últim hem utilitzat les imatges reals per fer el train negatiu.

3.2 Generació d'imatges sintètiques

Al inici del projecte no s'ha utilitzat cap imatge real completa. És a dir, per tal de poder fer proves, s'ha agafat diferents imatges reals i s'han anat retallant els dígit per separat manualment. Això dóna la possibilitat d'obtenir números de sèrie totalment diferents respecte el tamany, nitidesa, color, etc.



Un cop tenim una imatge creada amb el número de sèrie introduït per teclat inicialment, s'aplica el descriptor de contorn HOG. Això ens retorna un vector de característiques, és a dir, el vector que ens descriu la imatge.



Ara que ja tenim com preparar les imatges, necessitem fer un train i un test. Tant per les imatges de train com per les imatges de test, tindrem imatges positives i negatives. Les imatges positives són les que ens retorna el número de sèrie que hem introduït per teclat, i les negatives són números generats aleatòriament.

El procediment per fer el train, del quan obtindrem un model per després poder executar el test, és el següent:

Quan tenim unes quantes imatges amb el seu vector de característiques, les juntem en una matriu.

Tindrem una altra matriu d'etiquetes, que cada cop que s'actualitzi la matriu dels Hogs amb una imatge positiva, s'introduirà un 1, i si és una negativa, un 2.

De manera que queda de la següent manera:

Matriu de Hogs

Matriu Columna Etiquetes

P_1X_1	P_1X_2	...	P_1X_n	1
P_2X_1				1
...				1
P_nX_1	P_nX_n	1
N_1X_1	N_1X_2	...	N_1X_n	2
N_2X_1				2
...				2
N_nX_1	N_nX_n	2

P_i = Vector de característiques per cada imatge positiva "1".

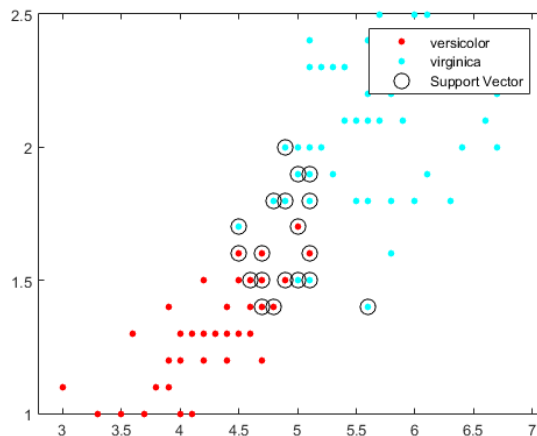
N_i = Vector de característiques per cada imatge negativa "1".

Cada fila és un vector i les columnes són els diferents valors del vector. Amb aquestes matrius de Hogs i d'etiquetes ja podem crear un model amb la funció `svmtrain(trHogs, labels)` de Matlab. (referència [10])

Per la part de test, s'aplicarà el mateix procediment que amb el train (amb el mateix número de sèrie introduït), però amb la diferència que cada vector de característiques en comptes de guardar-lo en una matriu, executarem el classificador SVM passant-li el mateix vector de característiques i el model com a paràmetres a la funció `svmclassify(model, featureVector)`. (referència [12])

El resultat obtingut serà de 1 o 2, depenent de què ha reconegut el test. En les últimes versions de Matlab ja estava a disposició una altra funció on deia amb quin percentatge de fiabilitat estava dins d'una classe o una altra. Al no tenir aquesta versió de matlab, només es disposa d'aquest classificador binari, sense saber si és una imatge molt clara en una classe o no.

Els resultats ideals serien que per totes les imatges de test positives ens retorni un 1 i per totes les negatives, un 2.



Aquest és un exemple de com seria el nostre classificador. Un color serien les instàncies positives i l'altre les negatives. Com es pot apreciar, hi ha una regió en el centre de la imatge que es mesclen punts dels diferents colors. Això és degut a que ha hagut instàncies del test que no ha tingut molt clar si és una imatge positiva o negativa i pot haver fallat.

3.3 Primers canvis en la metodologia inicial

Donat que les primeres proves amb imatges sintètiques donen un bon resultat, hem aplicat un canvi a l'hora de fer el test, ja que al tenir imatges totalment diferents les positives amb les negatives, ens ha reconegut perfectament i el que necessitem saber és si tenim un sistema que reconeix totes les classes perfectament o pot tenir errors.

Llavors a les imatges de test negatives, hem fet que en comptes de que agafi imatges totalment aleatòries, que agafi els mateixos números de sèrie de les imatges positives però aplicant 1, 2 o 3 canvis. Així es podrà veure com el classificador ja no actua tan bé. Aquest és un exemple de train positiu i el test negatiu (efectuant 3 canvis), i per tant, al no ser la mateixa imatge, el test ens hauria de retornar un 2, però en els resultats ja es veurà com no sempre el programa té clar que sigui negativa.



De imatges de train hem agafat 50 positives i 50 negatives. Per les de test, 50 positives i 500 negatives. Al crear-les nosaltres, podem fer-ne tantes com volguem sense necessitat de tenir-ne a la base de dades.

3.4 Primeres proves amb imatges reals

Les instàncies de train s'han continuat agafant les imatges sintètiques. No hem variat la quantitat de imatges tant de positives, com de negatives, 50.

En el test, ja s'han agafat de reals. De positives només hi ha una, ja que no tenim més imatges amb el mateix número de sèrie. Per les negatives hem agafat 50 imatges aleatòries dels altres comptadors.

3.5 Un nou mètode de testejar

Dividim les imatges en 2 carpetes, una per les de train, i les altres per les de test.

Un canvi efectuat és en el train negatiu. Les imatges ja no seran aleatòries, sinó que agafem els números de sèrie d'algunes imatges reals de la carpeta de train i obtenim la seva corresponent imatge sintètica. És a dir, si a la carpeta de train hi ha un arxiu "87654321.jpg" que es correspon a un comptador amb aquest número de sèrie, crearem una imatge sintètica amb el "87654321".



En el procediment del train positiu, només hem variat en quantes imatges positives hem generat, el primer train 50, el segon 25, i el tercer 100.

TRAIN 1	50 positives	Sintètiques
	50 negatives	Sintètiques

TRAIN 2	25 positives	Sintètiques
	25 negatives	Sintètiques

TRAIN 3	100 positives	Sintètiques
	100 negatives	Sintètiques

Per cada tipus de train, hem realitzat 50 tests. Per cada test hem agafat una imatge diferent com a positiva i les altres 49 com a negatives, de manera que les acabem utilitzant totes, ja que estan dins la carpeta de test, separades de la resta.

TEST	50 vegades	1 positiva	<i><u>POSITIVES OK</u></i> 50
		49 negatives	<i><u>NEGATIVES OK</u></i> <i>TOTAL(49x50vegades)</i>

Finalment, obtenim un threshold (%) on ens indica quin percentatge de positives i de negatives ha reconegut bé.

Les positives OK seran aquells tests positius que el classificador SVM ens ha retornat un 1 i les negatives OK són les que ens han retornat un 2. Al final, per una banda dividirem el número de positives que ha donat bé entre les 50 vegades que s'ha realitzat el test, i per l'altra les negatives que han donat bé entre totes les vegades que s'ha fet el test per les 49 imatges que hi ha.

3.6 Train amb imatges reals

Continuem avançant el projecte, aquest cop introduint un nou procediment de train. Com que de positives no podem obtenir 50, continuaran sent imatges sintètiques. Ara bé, les negatives ja seran reals, tot i que seran diferents a les agafades del test.

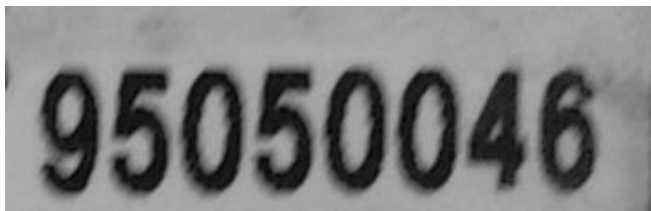
TRAIN 4	50 positives	Sintètiques
	50 negatives	Reals

El test serà el mateix que els mètodes anteriors. Més endavant veurem els resultats obtinguts, que no han sigut els esperats.

3.7 Modificació de les imatges reals

La següent modificació és afegir uns marges al voltant de la imatge sintètica, ja que en les fotografies de les imatges reals, el número no ocupa el 100% de la imatge, sempre hi ha un espai al voltant.

Una imatge real pot ser com aquesta:

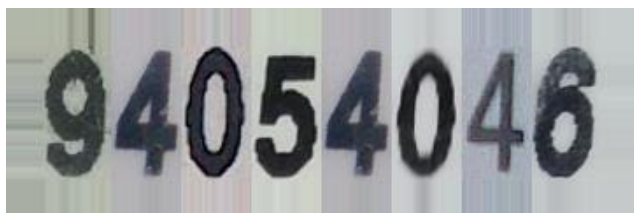


Lavors agafem el número de sèrie, creem la imatge sintètica, i al afegir-li el bord queda així:



De manera que el número ja queda més a l'interior de la imatge. Això farà que es reconeixi millor.

Si a més a més, es fa una altra modificació i aquest marge es posa del color de la imatge, al fer el HOG, els vectors no detectaran un quadre negre:



3.8 Nous marges en les imatges

Al veure que les imatges reals que tenim no sempre estan al centre de la imatge, hem modificat el nombre de píxels dels marges a l'hora de fer el train. S'ha fet de forma aleatòria per així a l'hora de entrenar el model, no sempre agafi la mateixa posició dels números, de manera que quan es fa el test, no hagi d'estar sempre al centre.

La funció que s'ha utilitzat de matlab és :

```
w=randi([5 40])
```

```
h = randi([5 40])
```

```
Img=padarray(imatgeAnterior,[w h], 'replicate','both');
```

La funció s'ha buscat a la referencia [1].

La “imatgeAnterior” és la imatge sense el marge. Els valors de w i h són els píxels, d’amplada i alçada respectivament, escollits de forma aleatòria entre 5 i 40. Com que afegim píxels a la imatge original, després s’ha de reduir la imatge al tamany de la original.

```
ImgFinal = imresize(Img, [115 360]);
```



Depenent del marge que es deixi, els números sortiran més grans o més petits, ja que al canviar la mida de l'arxiu a uns píxels determinats, aquests números s'estiren o es comprimeixen.

Per últim, per veure a partir de quantes imatges de train negatives es continua optimitzant els resultats, s'han anat fent proves de 50, 200, 500 i 1000. De train positives sempre han sigut 50.

4 PLANIFICACIÓ

4.1 Planificació inicial

[illegible]

Aquesta va ser la primera planificació que es va fer al inici del projecte. Més endavant, han anat sorgint detalls, problemes o petits canvis que han portat a una altra planificació.

Les primeres setmanes van ser per analitzar el projecte que hi havia sobre la validació del número de sèrie. Es va haver d'evaluar el projecte amb les alternatives a introduir. Necessàri també el coneixement dels mètodes que s'havien d'utilitzar, el HOG i el SVM.

A partir d'aquí ja era construir els mètodes, tot i que no sabia el que tardaria, es va plantejar d'aquesta manera.

4.2 Planificació final

[illegible]

Fase 5: Passar una sliding-window per la imatge - Aconseguir núm. identificador															
Fase 6: Validació de núm. utilitzant un "treshold (%)" i cercant a la base de dades															
Fase 7: Test de totes les imatges i tots els models de comptadors.															

Com es pot veure no s'ha modificat gaire la planificació inicial de la final. La fase 5 canvia una mica. No es fa el reconeixement de patró. En comptes d'utilitzar una sliding-window, el que es fa és utilitzar la mateixa imatge real i moure-la de posició perquè el model tingui més variabilitat.

També es fan proves amb el mateix model de comptador. Finalment s'introdueix una nova fase on es realitzen els tests genèrics.

4.3 Planificació inicial en hores

- Planificació del projecte: 10h
- Recerca d'informació i obtenció de coneixements ampliat: 10h
- Generació d'imatges: 15h
- Aplicació de HOG i comprovació amb funció hogdraw: 15h
- Entrenament SVM: 15h
- Separar instàncies positives i negatives dels diferents models: 20h
- HOG + SVM en imatges dels comptadors de gas: 35h
- Reconeixement de patró i utilització de Sliding-Window: 30h
- Validació del número obtenint el resultat més alt (treshold): 25h
- Redacció informes tècnics: 30h

4.4 Planificació final en hores

- Planificació del projecte: 10h
- Recerca d'informació i obtenció de coneixements ampliat: 10h
- Generació d'imatges: 20h
- Aplicació de HOG i comprovació amb funció hogdraw: 25h
- Entrenament SVM: 35h
- Separar instàncies positives i negatives dels diferents models: 20h
- HOG + SVM en imatges dels comptadors de gas: 35h
- Validació del número obtenint el resultat més alt (treshold): 30h
- Redacció informes tècnics: 50h
- Modificacions després de resultats finals: 30h

TOTAL HORES: 265h

5 RESULTATS

5.1 Resultats amb imatges sintètiques

Per cada train que fem, per exemple 50-50, que vol dir que fem 50 imatges positives i 50 imatges negatives, fem 50 de test, tant de positives com de negatives.

Train	Positives OK	Negatives OK
50-50	100%	100%
50-100	100%	100%

Les primeres proves que vaig fer per comprovar que funcionava el HOG i el classificador SVM van sortir bé.

Era bastant clar que si es creen les imatges sintètiques de la mateixa manera tant pel train com pel test, al final el classificador dona el resultat bo.

Vistos aquests resultats, es fa el canvi en el test negatiu i es posa casi igual que el train positiu, introduint 1, 2 o 3 canvis. S'escull una posició a modificar i cada test negatiu va canviant de número.

1 canvi:

Posició modificada pels tests negatius: "1 2 3 4 5 6 7 8"

Train	Positives OK	Negatives OK
50-50	100%	0%
50-100	100%	0%

2 canvis:

Posicions modificades pels tests negatius: "1 2 3 4 5 6 7 8"

Train	Positives OK	Negatives OK
50-50	100%	0%
50-100	100%	0%

3 canvis:

Posicions modificades pels tests negatius: "1 2 3 4 5 6 7 8"

Train	Positives OK	Negatives OK
50-50	100%	11/50
50-500	100%	30/50
50-1000	100%	31/50

Es pot veure que fent tant sols 1 o 2 canvis, el classificador no nota la diferència amb el train positiu i per tant, les negatives les posa a la classe 1, és a dir, com si fossin positives.

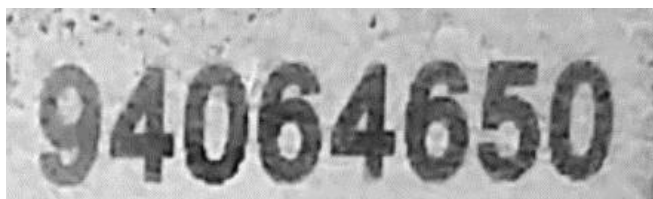
Al fer 3 canvis ja comença a notar la diferència. Al veure que amb 50 instàncies negatives de train ja reconeixia 11 bé, he pujat a 500 i m'ha donat un resultat millor, 30 de 50. Com que m'ha generat un bon canvi en els percentatges, he duplicat el train negatiu a 1000. Els resultats sorprenentment s'han mantingut, ja que el model ja no ha pogut entrenar millor.

5.2 Resultats amb canvis en el test

	POSITIVES	NEGATIVES
TRAIN	50	50
TEST	1	50

Test	Positives OK	Negatives OK
1-50	0%	43/50 = 86%

Com a imatge positiva he utilitzat aquesta:



El test positiu m'ha retornat un 2, per tant, no l'ha reconegut bé.

En canvi el test negatiu si que ha donat uns bons resultats amb les imatges de test reals. Un 86% ha retornat un 2.

Al no tenir un percentatge de fiabilitat de que un test pertanyi a una classe, no es pot saber si falla molt clarament o no.

5.3 Resultats amb les imatges de train negatiu diferents

El train continua sent imatges sintètiques però ara els números no són aleatoris. S'agafen de la carpeta de train.

El Test continua sent 1 positiu i 50 negatius. L'executem 50 vegades i calculem el treshold%.

Els resultats amb els 3 Trains diferents són:

Train	Positives OK	Negatives OK
25-25	34%	86,9%
50-50	30%	85,8%
100-100	32%	84,7%

Es pot apreciar que siguin quantes siguin les instàncies de train, el resultat no varia. El test positiu oscil·la sobre el 30% i el test negatiu sobre el 85%.

5.4 Resultats incorporant imatges reals en el train

Ara les imatges de train negatives són reals. Les positives es continuen creant sintèticament.

Train	Positives OK	Negatives OK
25-25	0%	100%
50-50	0%	100%
100-100	0%	100%

Com es pot veure, no ha funcionat molt bé de primeres. Al no reconèixer ninguna positiva ens ha fet mirar i comparar les 2 imatges, la sintètica i la seva real, i s'ha vist que es necessitava un marge, ja que el número no ocupa tot l'espai de la imatge.

5.5 Resultats incorporant un marge a les imatges reals

A les imatges sintètiques, les de train positiu, se li ha afegit un marge negre de 20 píxels.

Train	Positives OK	Negatives OK
25-25	8%	95,1%
50-50	6%	99%
100-100	2%	99,6%

Tot i que semblen resultats bastant baixos, a les conclusions s'explica que poden no ser-ho.

Ara els marges de 20 píxels són del color de la imatge.

Train	Positives OK	Negatives OK
25-25	52%	47,8%
50-50	50%	50,2%
100-100	54%	46,24%

Els resultats oscil·len en el 50% - 50%. A les conclusions explicaré per què aquests resultats no són els ideals.

Finalment, les últimes proves que s'han fet són variant el marge. Primer de forma aleatòria, i més tard, manualment posant un marge que oscil·li entre 5 i 40 píxels.

De forma aleatòria:

Train	Positives OK	Negatives OK
25-25	58%	42,9%,
50-50	48%	51,88%
100-100	46%	50,82%

Amb marge 5 a dalt i a baix, i 40 a esquerra i dreta:

Train	Positives OK	Negatives OK
50-50	26%	76%

Amb marge 10 a dalt i a baix, i 20 a esquerra i dreta:

Train	Positives OK	Negatives OK
50-50	0%	100%

Amb marge 40 a dalt i a baix, i 5 a esquerra i dreta:

Train	Positives OK	Negatives OK
50-50	10%	94,5%

Per últim, per veure si podia optimitzar els resultats, he agafat les imatges amb el marge de 5 a dalt i a baix, i 40 a esquerra i dreta, i he anat posant més imatges de train negatiu per tenir un millor model i així veure fins on milloren aquests resultats.

Train	Positives OK	Negatives OK
50-50	26%	76%
50-100	36%	65%
50-200	18%	84,8%
50-500	22%	82,3%

6 CONCLUSIONS

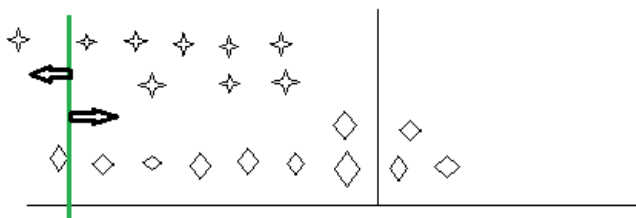
Després de tots els resultats obtinguts durant el projecte, es poden treure una sèrie de conclusions o hipòtesis bastant lògiques i raonables.

Per començar, quan s'ha treballat amb imatges sintètiques, el sistema ha funcionat perfectament i s'han tret uns grans resultats. Però, de fet, a la vida real no hi ha fotografies sintètiques, així que els resultats verdaderament importants són els fets amb proves reals.

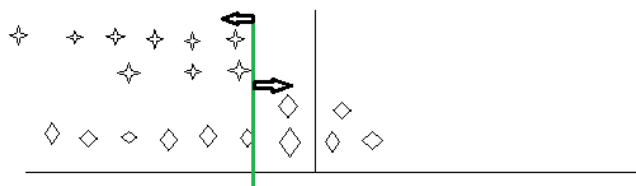
Quan s'ha trobat la manera de que les imatges sintètiques s'assemblassin a les reals, és a dir, afegint un marge, han sorgit varis detalls interessants. Alguns percentatges han oscil·lat sobre el 50%-50% i altres més cap el 0%-100%.

Doncs bé, he comentat anteriorment que serien més ideals els resultats de 0% i 100%, ja que el nostre classificador és binari totalment. És a dir, no ens retorna el threshold (%) del qual el sistema està segur que una imatge pertany a una classe o una altra.

Un exemple de 0% i 100% aproximadament seria aquest:

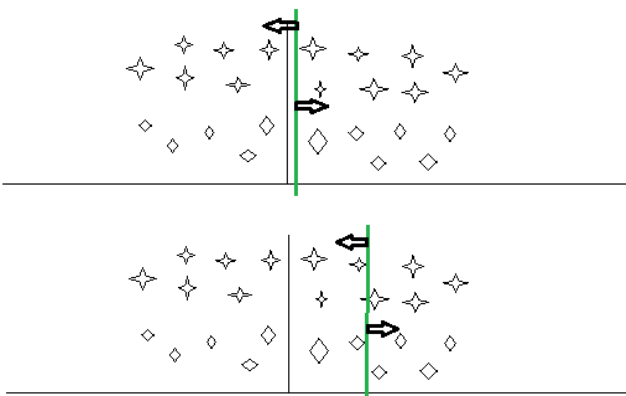


Les estrelles són les imatges de test que les ha reconegut com positives i els rombes les que ha reconegut com a negatives. Es pot veure que cap a l'esquerra només agafa una estrella i que de rombes quasi bé tots. Doncs bé, si podem moure aquesta línia separatòria, el barem, fins agafar-ne totes les positives, encara agafaríem bastantes negatives. És a dir, quedaria així:



Aquest barem el podríem moure si tinguéssim el percentatge de seguretat que una instància sigui d'una classe o una altra. Es podria anar jugant amb el resultat per optimitzar els valors finals.

En canvi, si tenim uns resultats de 50%-50%, tindríem el problema que no podríem jugar amb aquest barem, ja que encara que la féssim moure cap a un costat o cap a l'altre, sempre treurem el mateix percentatge que guanyarem en l'altra classe:



Si movem el barem cap a la dreta (és a dir, agafem més positives), perdem la mateixa quantitat de negatives. Per tant, sempre tindrem un dels dos valors baix.

7 TREBALLS FUTURS

Per poder millorar aquest projecte, penso que el més important és poder jugar amb el barem per poder obtenir resultats més òptims. Per a fer-ho, es necessita una versió de Matlab superior a la del 2013, ja que hi ha noves funcionalitats que ens serveixen.

Una altra possibilitat per millorar el projecte podria ser utilitzar el sistema operatiu Linux, ja que deixa executar arxius MEX. Això s'ha necessitat a meitat projecte per poder reutilitzar el model d'entrenament anterior. (referències [5], [8] i [9])

També és recomanable trobar més solucions a la hora de crear les imatges sintètiques, poder fer-les més semblants a la realitat.

Per últim, seria adequat partir d'una base amb un bon material per fer aquestes proves, sense tenir quasi bé totes les imatges reals amb soroll, borroses, etc. Això ha sigut un inconvenient a l'hora d'obtenir els resultats.

8 AGRAÏMENTS

Agraeixo tot el seguiment realitzat pel meu tutor de projecte, Ernest Valveny, per encaminar-me en tot moment quins passos eren els més adients durant el curs. He estat en continu contacte amb ell per solucionar tots els problemes trobats després d'una hipòtesis inicial.

També agrair el suport rebut per part de la meva família durant la realització de tot el projecte.

BIBLIOGRAFIA

- [1] Funció per afegir marges a les imatges : (06/2015)
<http://es.mathworks.com/help/images/ref/padarray.html>
- [2] Explicació d'un Reconeixement òptic de caràcters: (03/2015)
<http://es.abbyy.com/finereader/ocr/>
- [3] Informació HOG: (04/2015)
<https://software.intel.com/en-us/node/529070>
- [4] Jon Almazán, Albert Gordo, Alicia Fornés, Ernest Valveny;
Segmentation-Free Word Spotting with Exemplar SVMs
Departament ciències de la Computació - UAB (Bella-terra)
- [5] Ajuda en matlab. Per utilitzar funció MEX per compilar un arxiu C. (04/2015)
<http://www.mathworks.es/help> (20/02/2015)
- [6] Informació sobre el SVM: (04/2015)
http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm
m (05/03/2015)
- [7] Funcionalitat de les finestres lliscants: (03/2015)
http://cs.brown.edu/courses/cs143/proj4/Face_detector_with_sliding-window (03/2015)
- [8] Creació de MEX files : (04/2015)
<http://es.mathworks.com/help/matlab/ref/mex.html#bt2l4qg-3>
- [9] Problemes a l'hora de compilar un arxiu .C amb matlab: (05/2015)
<https://github.com/lloyd/node-toobusy/pull/7>
- [10] Funció per obtenir el model: (04/2015)
<http://es.mathworks.com/help/stats/svmtrain.html>
- [11] Com s'utilitza el SVM: (05/2015)
<https://www.youtube.com/watch?v=4GB5HVw0CuM>
- [12] Funció per determinar a quina classe pertany una instància: (05/2015)
<http://es.mathworks.com/help/stats/svmclassify.html>
- [13] Explicació mètode HOG: (06/2015)
<http://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/>